Image Processing based Image to Cartoon Generation: Reducing complexity of large computation arising from Deep Learning

Shruti Kumari Shrivastava
Electronics and Communication Engineering Department
Institute of Technology, Nirma University
Ahmedabad – 382481, Gujarat, India
shrutishrivastava22ss@gmail.com

Ruchi Gajjar
Electronics and Communication Engineering Department
Institute of Technology, Nirma University
Ahmedabad – 382481, Gujarat, India
ruchi.gajjar@nirmauni.ac.in

2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES) | 979-8-3503-2391-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/CISES58720.2023.10183524

***Abstract*— This paper proposes an approach to convert real life images into cartoon images using image processing. The cartoon images have sharp edges, reduced colour quantity compared to the original image, and smooth colour regions. With the rapid advancement in artificial intelligence, recently deep learning methods have been developed for image to cartoon generation. Most of these methods perform extremely huge computations and require large datasets and are time consuming, unlike traditional image processing which involves direct manipulation on the input images. In this paper, we have developed an image processing based method for image to cartoon generation. Here, we perform parallel operations of enhancing the edges and quantizing the colour. The edges are extracted and dilated to highlight them in the output colour image. For colour quantization, the colours are assigned based on proposed formulation on separate colour channels. Later, these images are combined and the highlighted edges are added to generate the cartoon image. The generated images are compared with existing image processing approaches and deep learning based methods. From the experimental results, it is evident that the proposed approach generates high quality cartoon images which are visually appealing, have superior contrast and are able to preserve the contextual information at lower computational cost.**

Keywords*— *Canny edge detector, Median filter, Dilation, Image to Cartoon, Colour quantization

I. INTRODUCTION

As the digital world is achieving great heights, people are looking for different ways to represent themselves. One of the best ways is creating their cartoon images. People are using wonderful platforms like Avatoon, PhotoLab and many more. Even one of the popular platforms, Snapchat, is providing the cartoon images. Some big and popular applications like Photoshop, Adobe Illustrator, and many others help in achieving the goal of image conversion to cartoon. Working on thousands of images, it will be very difficult to implement the process individually on these platforms. Thus, it is required that we adopt an efficient algorithm that can skip the human involvement and within milliseconds can produce the cartoon image.

The task of converting an image into a cartoon can be achieved with Image Processing as well as Machine Learning. There are many existing algorithms reported in literature for this. The image processing based techniques use filtering and morphological operations for cartoon image generation. Machine learning approaches involve utilization of complex architectures such as GANs for image to cartoon conversion. These Machine Learning algorithms are time consuming and require a huge dataset for training purposes. Sometimes, the model is not always the desired one and may lead to underfitting or overfitting. Therefore, with fundamental concepts of Image Processing, the time consumption, complex computations and model creation can be avoided.

The objective of the paper is to develop an algorithm to convert the image into a cartoon image. The said problem statement is addressed by applying the concept of median filter, edge detection, dilation, average filter, quantization, and image channel in MATLAB. The input image is a 3D matrix that contains rows, columns, and three channels (RGB). All the mentioned procedures are individually applied on these channels and then combined to give the desired output. There may be a chance of the addition of additive noise. Such noise can be overcome by applying median filters and average filters. For detecting the edges, a Canny edge detector is used, which is further dilated. This operation is applied on the image by converting it to grayscale image. For reducing the colour quantity, every pixel value of the image is quantized. After performing both procedures, the detected edge is combined with the output of colour saturation and quantization process. The final image gives the cartoon version of the input image.

II. LITERATURE SURVEY

The image to cartoon conversion involves thickening and highlighting the edges, and reducing the colour quantity. The algorithm proposed by [1] detects the edges by applying a Median filter followed by the canny edge detector. Thereafter, the morphological operations are applied to thicken and smoothen the contours of edges. In [2], edges are detected by the canny edge detector after applying the bilateral filter. Further, erosion is done to thicken the detected edge. The cartoon images have bold edges to separate different areas. Therefore, edge detection is required. The edge detection will result in thin lines which will be thickened by morphological operations.

There are many edge detection algorithms amongst which canny edge detectors are widely used. [6] compared Sobel edge detector and canny edge detector through FPGA implementation. The result [6] showed that the Sobel edge detector showed more thin edges compared to the canny edge detector. Apart from RGB channels, the edges can be detected in YCbCr channels too. [7] proposed the method of detecting the edges by transforming the image into YCbCr channel, where Y channel is extracted to display the edges. Four edge detectors are selected, Prewitt, Sobel, Roberts and Canny, to operate on image. By analyzing and comparing the results obtained, the canny edge detector gives better image edge amongst rest detectors. The performance parameters can give the comparison analysis for different operators. [8] gives the comparative analysis of performance parameters, PSNR (Peak Signal to Noise Ratio), MSE (Mean Square Error), MAXERR (Maximum Squared Error), L2RAT (ratio of squared norm of signal to the input signal), and processing time, of various edge detectors. Upon calculating these parameters, the canny edge detector is selected as the optimal edge detector [8].

Once edges are detected, the colour quantity is reduced. [1] focuses on colour quantization, where bilateral filters and median filters are applied. The pixel values are quantized by dividing the pixel value with a fixed value, the floor of the obtained number is taken and then multiplied with the same fixed value. Once the final quantized image is obtained, it is combined with the edges and final output gives the desired result. However, the algorithm is bad at handling portrait images. The colour quantization can be performed on different colour models. For reducing the colour quantity, in [2], the image is converted to HSV model and then k-means clustering is applied. The contours are drawn on image. After this, erosion is done to resemble it to human painting. And finally, the image is again converted back to RGB and the output image is the cartoon image. The output given in [1] and [2] are purely based on Image Processing. The cartoon image can be obtained by the concept of vector quantization also. This gives the painterly effects on image. [9] uses LBG (Linde-Buzo-Gray), KPE (Kekre's Proportionate Error), and KMCG (Kekre's Median Codebook Generation) algorithms to obtain cartoon images having painterly effects.

For creating cartoon style image from real world photos, [11] proposed CartoonGAN, Generative Adversarial Network (GAN) framework. This proposed method takes unpaired photos and cartoon images for training. [11] compared CartoonGAN with CycleGAN, a CNN-based stylization, and observed that CartoonGAN takes significantly much less training time. It is also given that for each epoch, CycleGAN and CycleGAN with Lidentity take 2291.77s = 38.2 minutes and 3020.31s = 50.34 minutes respectively. And CartoonGAN takes 1517.69s = 25.29 minutes. The image can be converted to cartoon image using Machine Learning Methods also. [3] involves Neural Style Transfer, a Machine Learning method, which requires two images i.e., the first is the original image and the second is the styled image. The styled image is generally artistic image that can be applied to the original image. Colour reduction algorithms can be proposed by manipulating the histogram of the image too. [10] proposed colour reduction algorithm based on analysis of the colour histogram. According to the algorithm, the histogram of the original image is divided into regions as per the colours in the new image. The colour with maximum pixels in each region is considered as base colour to the new palette.

III. Proposed Methodology

The image contains a lot of information along with noise. Thus, the noise present in the image must be eliminated such that the information should not change. Therefore, initially, to eliminate the noise, the Median filter is applied. Fig. 1 shows the block diagram of the proposed method. The image to cartoon conversion requires extracting edges and reducing the colour quantity. The edge extraction and reducing the colour quantity procedure are different and independent from each other. Either they can be simultaneously performed, or can be implemented one after another. For extracting the edges from the image, it is converted to grayscale image and edge detection and morphological operations are applied. For reducing the colour quantity, a group of pixel values will be mapped to a particular pixel value. This process will be a point processing method. Once both are implemented, they are combined logically. Wherever, there will be an edge, black pixel value will be assigned to the image obtained by colour quantity reducing process.

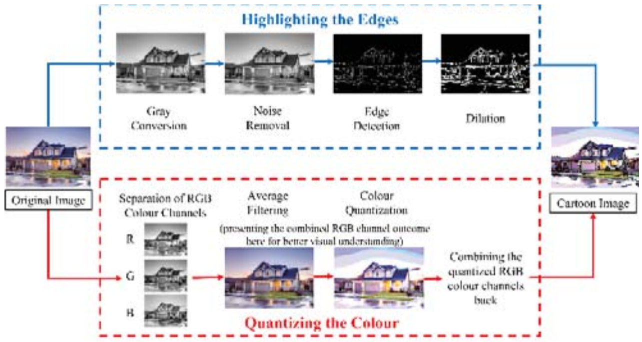


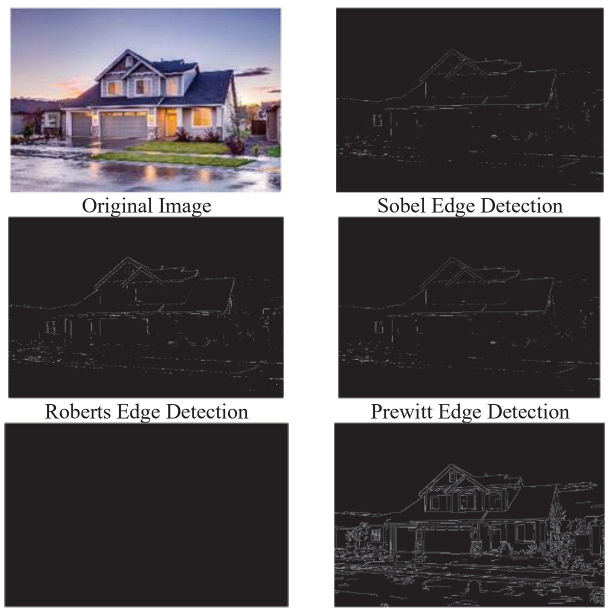
Fig. 1. Block diagram of proposed method

A. Noise removal

The Median filter is used to remove any noise present in the image. The Median filter will remove any salt and pepper noise that is present without much blurring, thus preserving the edges, that is to be detected. The Average filter can cause blurring; therefore, it will be difficult to detect the edges.

B. Edge detection

After the noise is removed, the edge from the image is detected by using edge detectors. In order to find the precise edge detector, we implemented various edge detectors on a sample image keeping a threshold value of 0.2. Fig. 2 presents the outcomes of these edge detectors. It can be observed that Canny edge detector gives the better edges than Sobel, LoG, Roberts, and Prewitt edge detectors, while LoG is not able to detect edges at the set threshold. Thus, the Canny edge detector is used in our proposed method.



LoG Edge Detector Canny Edge Detector

Fig. 2. Different edge detectors output for threshold = 0.2 with original image adapted from [5]

C. Dilation

Once the edge is detected by the canny edge detector, the edges must be enhanced. In cartoons, they have well defined edges unlike real life people. Hence, the morphological method, dilation, is applied to get the thicker and bold edge. The structuring element used is of order line which is at angle 0o and 90o. After dilation, the edge clearly differentiates different regions. This image is a binary image where 1 represents white, the edges, and 0 represents black, the background. In the final image, wherever the pixel value will be true, i.e., 1, there will be the black edge.

D. Averaging Filter

The Averaging filter is the smoothing filter that causes blur effect on images. This filter is applied on the three channels of the original image. The aim of using this filter is to blur out the contents so that every pixel value gets influenced by their neighbour pixels which will reduce the colour quantities.

E. Colour Quantization

The cartoon image always has a less colour palette compared to its original image. Therefore, to reduce the number of colour pixels in the image, each current pixel value, p, of the image is divided by a constant (a-b), where a and b are constant values, then converted to the floor value of the result obtained. The final pixel value, P, assigned will be the multiplication of integer obtained by taking floor value and $(a+\frac{b}{2})$.

$$P = \left\lfloor \frac{p}{a-b} \right\rfloor * \left(a + \frac{b}{2}\right) \tag{1}$$

In order to reduce the colour quantity, a group of pixel values are mapped to a particular pixel value. This is performed using eq. (1). For an 8-bit integer number, the total pixel levels for one channel will be 256. Upon diving it by (a-b), the colour quantity reduces to $[\frac{256}{(a-b)}]$. Thus, for RGB image, the colour quantity will be reduced to $[\frac{256}{(a-b)}]3$ [1]. For example, when a = 50, b = 10, and p = 166, the new pixel value will be 220. Therefore, the 166-pixel value is mapped to 220. Table 1 shows how the group of pixel values are mapped to particular pixel values for a = 50 and b = 10. Thus, the total quantity of colour will be $[\frac{256}{(a-b)}] = [\frac{256}{(50-10)}] = [6.4] = 6$ for one channel. Therefore, for 256 different pixel values, we are reducing it to only 6-pixel values. These pixel values will have different effects in different channels.

TABLE I. PIXEL VALUES MAPPING

Sr. No.	*Current Pixel Values*	*Mapped Pixel Values*
1	0-39	0
2	40-79	55
3	80-119	110
4	120-159	165
5	160-199	220
6	200-255	255

The 256-pixel values for a grayscale image are mapped to only 6-pixel values after quantization. The RGB image has three channels. Therefore, the mapped levels will have 6x6x6 different combinations of pixel values. Therefore, 256x256x256 = 1,67,77,216 combinations of levels are mapped to 6x6x6 = 216 combinations of pixel values.

F. Combining Edges and Colour Quantized Image

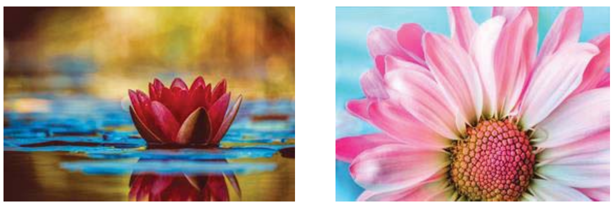
Once the edges are detected, and the quantized image is obtained for a particular value of a, the cartoon image can be obtained. For different channels, there are different quantized images that need to be converted back to a 3D matrix. Further, the edges which are detected should be added to the obtained 3D matrix to highlight the edges. Wherever the edges are detected, that pixel value is set to 0 in the 3D matrix to show it in black colour. Therefore, after quantization, the colour quantity in the cartoon image will be very less compared to the original image.

Fig. 3 shows the original images, quantized images and their respective histograms for three different channels. The histogram of original images has almost all the pixel values from 0 to 255 and are unevenly distributed. The histogram of the quantized image has only six different pixel values and are almost uniform. The quantized image is obtained using the eq. (1) where the original pixel value is divided by (a-b), the floor of the value obtained is then multiplied with $(a+\frac{b}{2})$ to get the mapped pixel value. By this equation, a range of histograms is mapped to a particular pixel value. This divides the whole image into different regions based on the pixel values. Thus, in the histogram of the quantized image, there are only 6 different pixel values for every channel.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

The algorithm described efficiently converts the image into its respective cartoon version. The objective is met by detecting edges using a Canny edge detector, Average filtering, and colour quantization. The variation in the values of a and b gives different colour quantities. The threshold values for edge detection can be varied based on the edge requirement such that the image appears to have edges.

In order to observe the effects of variation of the parameters a and b, as formulated in the proposed method, on the generated output cartoon image, a series of simulations were carried out for different values of a and b. As evident from eq. (1), the (a-b) controls the number of colour quantization levels and the value of a has to be greater than b. We varied a from 50 to 200, in a step size of 50 and b from 5, 10, 20 and 40. The results for a set of sample images are presented in Fig. 5, applied on original image of Fig. 4. Analyzing these images and comparing the generated images, it can be observed that for a=50 and b=10, the cartoon image has all the colours that are sufficient to represent the image.



(a) (b)

Fig. 3. Original Images for analyzing the effects of parameters *a* and *b*, adapted from [5]

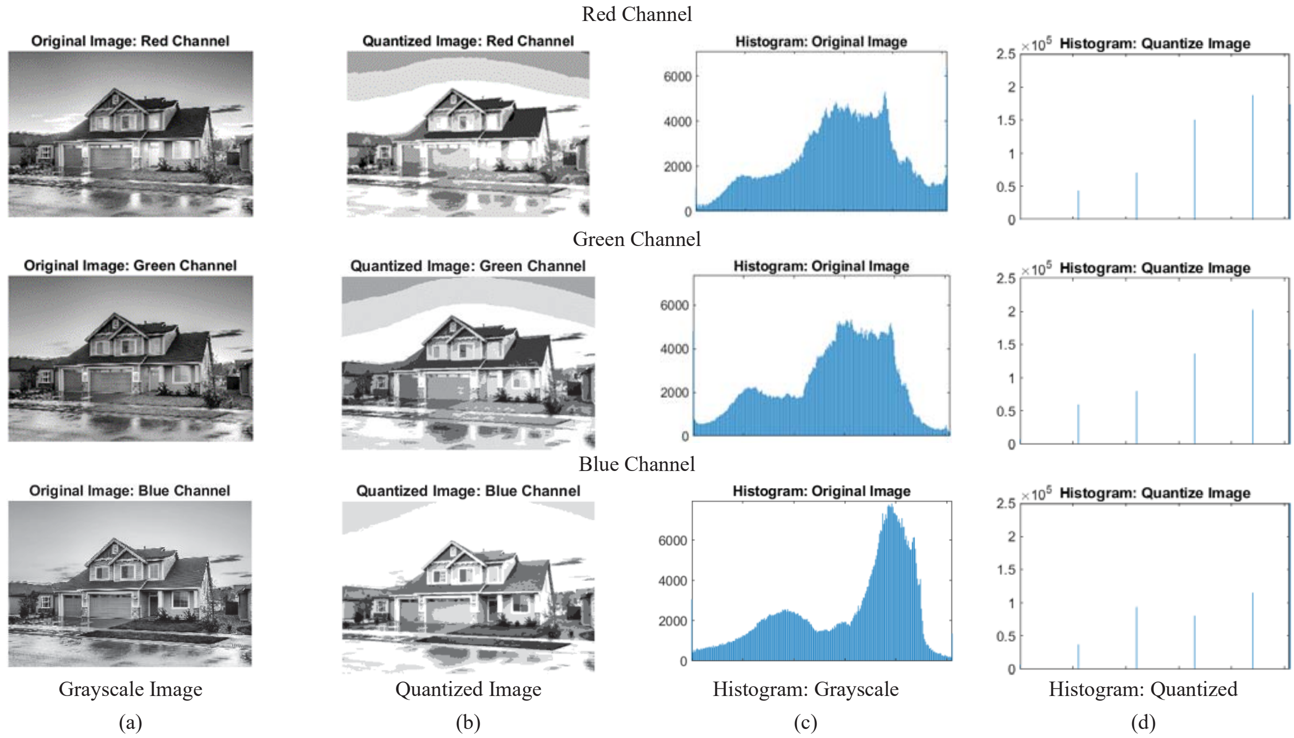


Fig. 4. Histogram of original image and quantized image for three channels. (a) Original grayscale Image, (b) Quantized Image (c) Histogram of original image, (d) Histogram of quantized image

Fig. 5 (a) and (b) gives the comparison between the cartoon image obtained for different values of a and b. For comparison purposes, two images are selected such that one is low contrast and the other is high contrast image. For both the outputs, it is observed that as the b value increases, keeping the value of a constant, the image becomes overexposed. The histogram shifts towards the right, to higher pixel values. And as the value of a increases, keeping the value of b constant, the image becomes underexposed. The histogram shifts towards the left side, to lower pixel values. Therefore, for the values of a and b, that gives higher difference, leads to a smaller number of pixel values. When the difference is 100, the total colour levels will be [255/100] = 2 and when the difference is 10, the total colour levels will be [255/10] = 25.

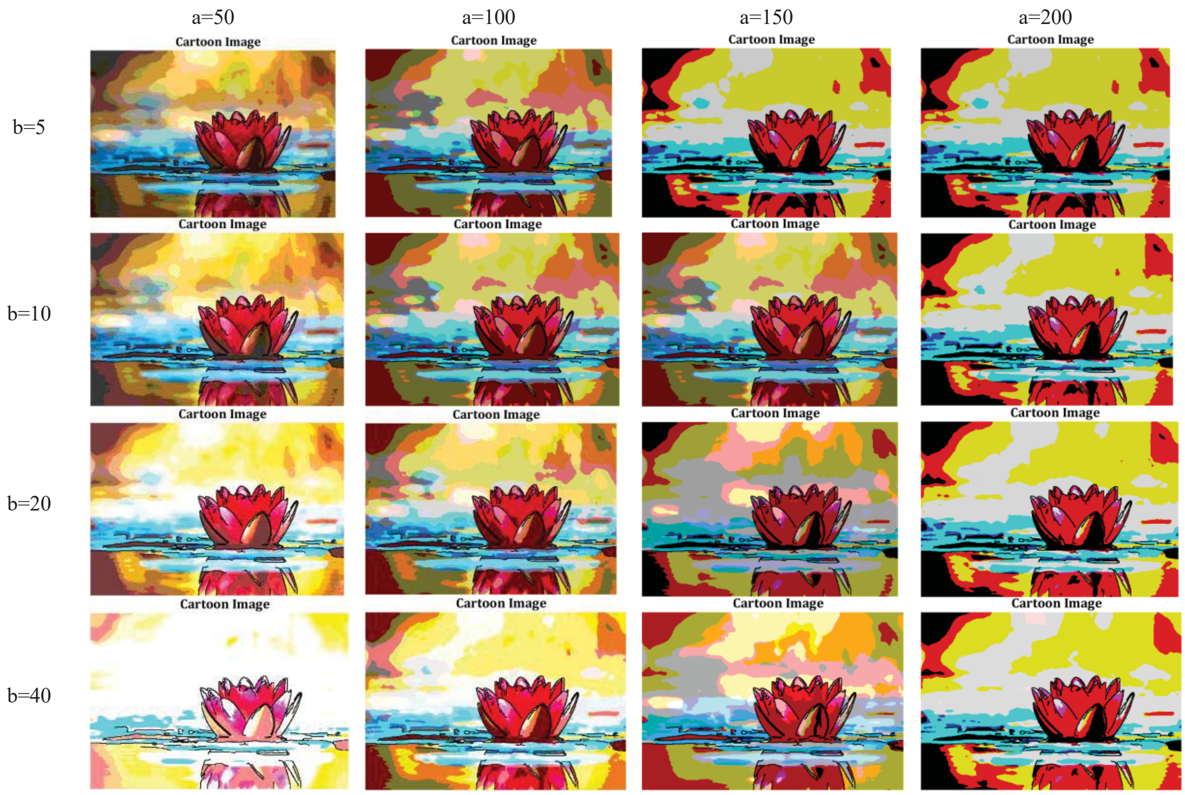
Cartoon images have limited colour palette and limited colour variation. More the number of colour levels, the more realistic the image will appear. Cartoon images are not intended to be realistic. It should have clear and bold edges, simple shapes and limited colour palette. Therefore, an image (grayscale) with 6-8 colour levels is sufficient to represent its cartoon image. With these statements and the results shown in Fig. 5, for a=50 and b=10, for both the images, it gives the minimum number of pixel values that are sufficient to represent the image as a cartoon.

To check the robustness of the proposed approach with the selected parameters, we generated cartoon images of some more sample images. Fig. 6 shows the results of cartoon images generated by the proposed method using a = 50 and b = 10. Fig. 6 (a) to (e) are original images from an open-source platform [5] and their respective cartoon images are presented in Fig. 6 (f) to (j). The generated cartoon images are dependent and a function of the actual colour encoding of the original image. It can be observed from Fig. 6, that our approach generates effective cartoon images for input (a) to (d). However, it doesn't give better results for image (e). Given that the image (e) has limited colours, white, gray and baby pink, the colour encoding them is near of white. So, the outcome is dominated by the edge detection, rather than colour quantization.

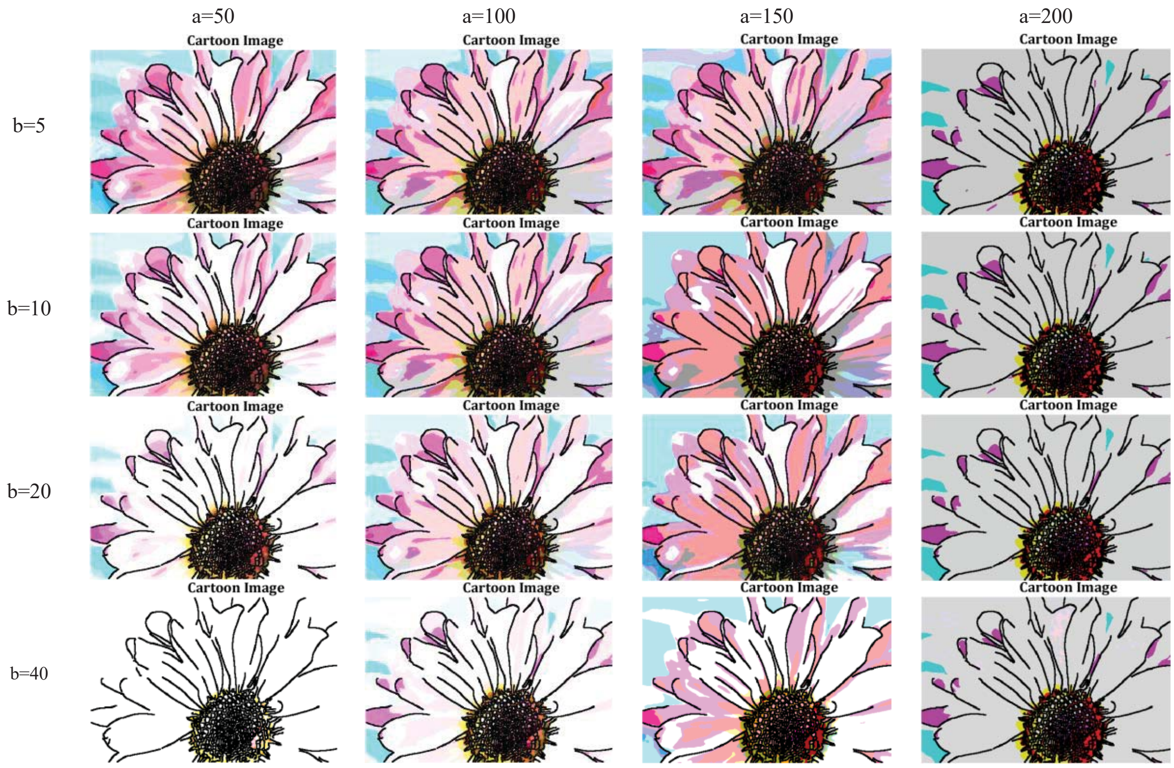
The results of the proposed method are also compared with some image to cartoon generation methods reported in literature. Fig. 7 gives the output from the algorithm adopted in [1] and [2] and the algorithm proposed. It is evident that the output produced by the proposed method gives enhanced output compared to rest. The edges are clearly visible and are able to separate the different parts of the image. The cartoon images generated by the proposed method exhibit high contrast in comparison to the other existing approaches.

Fig. 8 presents the outcome of the cartoon image generated by proposed method and the results of cartoon image generation using deep learning-based approaches presented in [13]. These methods employ GAN approaches. CartoonGAN [11] approach included convolutional neural networks with GANs. The visual observation of these results demonstrate that the images generated by GANs are accurate, but appear as synthetic data, in comparison to the image generated by proposed method, which gives a higher contrast and better cartoon effect.

These GAN based methods require trainings and hence it makes it time consuming. The time required by our method to generate a cartoon image is nearly to 0.7s. whereas the training time for deep learning-based approaches are very large. Specifically, the CartoonGAN [11] reported around 1517.69s per epoch, which is typically high as compared to traditional image processing.



(a) Output of proposed method with different parameter values, *a* and *b*, applied on Fig. 4(a)



(b) Output of proposed method with different parameter values, *a* and *b,* applied on Fig. 4(b)

Fig. 5. Cartoon images generated by proposed method with different parameter values, a and b, applied on Original Images of Fig. 4

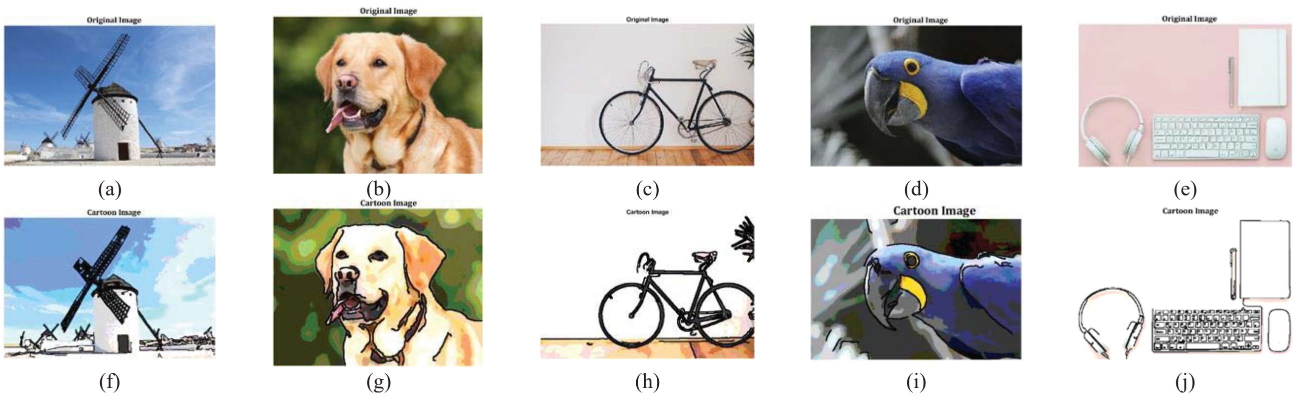


Fig. 6. Result of the proposed method. (a) to (e) Original images from [5], (f) to (j) generated cartoon images using a = 50 and b = 10.

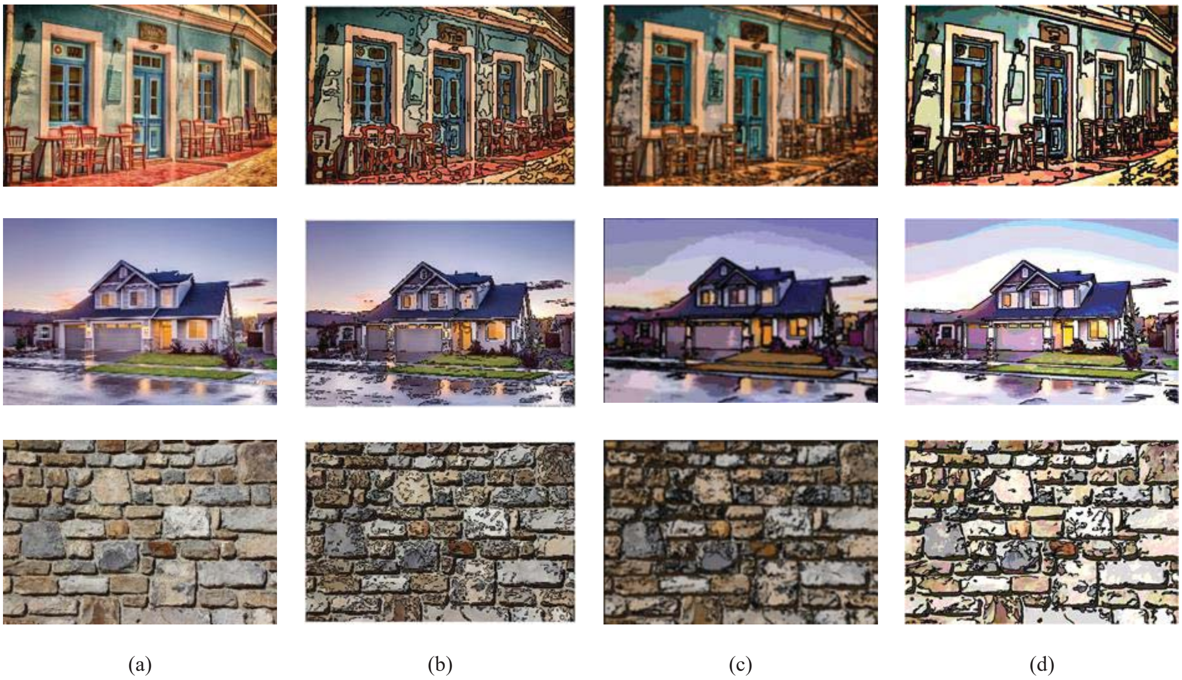


Fig. 7. Comparison of the cartoon images generated by the proposed method with existing Image processing-based approaches. (a) Original image, (b) Cartoon Image generated from [1], (c) Cartoon Image from [2] and (d) Cartoon Image from proposed method

Fig. 8. Comparison of the cartoon images generated by the proposed method with the existing GAN-based approaches presented in [13]

V. Conclusion

The image to cartoon generation method presented in this paper generates cartoon images using image processing operations. The goal is achieved by proposing a colour quantization method, which not only equalizes the histogram, but also enhances the dynamic range of the image. The output image is combined with thick and clear edges generated by performing edge detection and dilation after removal of the noise. The experimental results show that this algorithm is capable of producing the high quality, visually superior cartoon image from real-life images. The generated images exhibit better contrast and colour quantization in comparison to existing image processing based methods. It also performs better in comparison to deep learning based approaches, at lesser computations and processing time, eliminating the need for model training. This work can be further applied on video sequences for cartoon video generation.

References

[1] Kevin Dade, “Toonify: Cartoon Photo Effect Application”

[2] S. Wang and J. Qi, "A Novel Image Cartoonization Algorithm without Deep Learning," CIBDA 2022; 3rd International Conference on Computer Information and Big Data Applications, Wuhan, China, 2022, pp. 1-4.

[3] Chinmay Joshi, Devendra Jaiswal and Akshata Patil, “Technical Paper Presentation on Application of Cartoon like effects to Actual Images”, International Journal of Innovative Science and Research Technology, ISSN No: 2456-2165, Volume 4, Issue 3, March 2019

[4] Vaishali Sudarshan and Amritesh Singh, “Cartooning an Image Using OpenCV and Python”, ISSN: 2456-236X, Volume 4, Issue 2, 2020

[5] Images available at: Pixabay, https://pixabay.com/

[6] Kumar, S., Sagar, V. & Punetha, D. A comparative study on facial expression recognition using local binary patterns, convolutional neural network and frequency neural network. Multimed Tools Appl (2023). https://doi.org/10.1007/s11042-023-14753-y

[7] J. Dong, J. He and H. Wang, "Edge Detection of Human Face," 2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI), Fuzhou, China, 2021, pp. 596-601, doi: 10.1109/CEI52496.2021.9574525.

[8] A. Jose, K. Deepa Merlin Dixon, N. Joseph, E. S. George and V. Anjitha, "Performance study of edge detection operators," 2014 International Conference on Embedded Systems (ICES), Coimbatore, India, 2014, pp. 7-11, doi: 10.1109/EmbeddedSys.2014.6953040.

[9] A. B. Patankar, P. A. Kubde and A. Karia, "Image cartoonization methods," 2016 International Conference on Computing Communication Control and automation (ICCUBEA), Pune, India, 2016, pp. 1-7, doi: 10.1109/ICCUBEA.2016.7860045.

[10] S. O. Abter and N. A. Z. Abdullah, "An efficient colour quantization using colour histogram," 2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT), Baghdad, Iraq, 2017, pp. 13-17, doi: 10.1109/NTICT.2017.7976153.

[11] Y. Chen, Y. -K. Lai and Y. -J. Liu, "CartoonGAN: Generative Adversarial Networks for Photo Cartoonization," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 9465-9474, doi: 10.1109/CVPR.2018.00986.

[12] A. P. Singh, S. Kumar, A. Kumar and M. Usama, "Machine Learning based Intrusion Detection System for Minority Attacks Classification," 2022 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES), 2022, pp. 256-261, doi: 10.1109/CISES54857.2022.9844381

[13] F. Andersson and S. Arvidsson, “Generative Adversarial Networks for photo to Hayao Miyazaki style cartoons”. arXiv, 2020. doi: 10.48550/arXiv.2005.07702